

EPSON ESCPOS プロバイダ

Version 1.0.0

ユーザーズ ガイド

August 22 2017

【備考】

【改版履歴】

版数	日付	内容
1.0.0	2017-8-22	初版.

目次

1. はじめに.....	6
2. プロバイダの概要	7
2.1. 概要	7
2.2. インストール	7
2.3. メソッド・プロパティ	8
2.3.1. CaoWorkspace::AddController メソッド.....	8
2.3.1.1. Conn オプション	9
2.3.2. CaoController::Execute メソッド	9
2.3.3. CaoController::AddVariable メソッド.....	9
3. コマンド一覧	11
3.1. 印字命令コマンド	13
3.1.1. CaoController::Execute("PrintLF") コマンド	13
3.1.2. CaoController::Execute("PrintFeedPaper") コマンド	13
3.1.3. CaoController::Execute("PrintFeedLine") コマンド	13
3.2. 改行量コマンド	14
3.2.1. CaoController::Execute("ResetLineSpacing") コマンド	14
3.2.2. CaoController::Execute("SetLineSpacing") コマンド	14
3.3. 印字文字コマンド	14
3.3.1. CaoController::Execute("SetRightSideCharacterSpacing") コマンド.....	14
3.3.2. CaoController::Execute("SetPrintMode") コマンド	15
3.3.3. CaoController::Execute("SetUnderLineMode") コマンド.....	15
3.3.4. CaoController::Execute("SetEmphasizedMode") コマンド	16
3.3.5. CaoController::Execute("SetDoubleStrikeMode") コマンド	16
3.3.6. CaoController::Execute("SetFont") コマンド	17
3.3.7. CaoController::Execute("SetInternationalCharacter") コマンド	17
3.3.8. CaoController::Execute("SetRightRotate") コマンド	18
3.3.9. CaoController::Execute("SetCharacterCodeTable") コマンド	19
3.3.10. CaoController::Execute("SetSMModeUpsideDownMode") コマンド	20
3.3.11. CaoController::Execute("SetFontSize") コマンド	21
3.3.12. CaoController::Execute("SetWhiteBlackReverseMode") コマンド.....	21
3.3.13. CaoController::Execute("SetSmoothingMode") コマンド	22
3.4. 印字用紙コマンド	22

3.4.1. CaoController::Execute("GetPositioningInfo") コマンド	22
3.4.2. CaoController::Execute("FeedPaperToPeelingPosition") コマンド	23
3.4.3. CaoController::Execute("FeedPaperToCuttingPosition") コマンド	24
3.4.4. CaoController::Execute("FeedPaperToStartingPosition") コマンド	24
3.5. 印字位置コマンド	25
3.5.1. CaoController::Execute("MoveHT") コマンド	25
3.5.2. CaoController::Execute("MoveA") コマンド	25
3.5.3. CaoController::Execute("SetHTPosition") コマンド	26
3.5.4. CaoController::Execute("MoveR") コマンド	26
3.5.5. CaoController::Execute("SetSModeAlignment") コマンド	27
3.5.6. CaoController::Execute("SetSModeLeftMargin") コマンド	27
3.5.7. CaoController::Execute("MoveSModeBeginningLine") コマンド	28
3.5.8. CaoController::Execute("SetSModePrintAreaWidth") コマンド	28
3.6. メカコントロールコマンド	29
3.6.1. CaoController::Execute("CutPaper") コマンド	29
3.7. 漢字制御コマンド	30
3.7.1. CaoController::Execute("SetKanjiSetting") コマンド	30
3.7.2. CaoController::Execute("StartKanjiMode") コマンド	30
3.7.3. CaoController::Execute("SetKanjiUnderLineMode") コマンド	31
3.7.4. CaoController::Execute("StopKanjiMode") コマンド	31
3.7.5. CaoController::Execute("SetKanjiCodeSystem") コマンド	31
3.7.6. CaoController::Execute("SetKanjiSpacing") コマンド	32
3.7.7. CaoController::Execute("SetKanjiQuadrupleSizeMode") コマンド	32
3.7.8. CaoController::Execute("SetKanjiFont") コマンド	33
3.8. 補助機能コマンド	33
3.8.1. CaoController::Execute("RTMPowerOff") コマンド	33
3.8.2. CaoController::Execute("RTMClearBuffer") コマンド	34
3.8.3. CaoController::Execute("InitializePrinter") コマンド	34
3.8.4. CaoController::Execute("TestPrint") コマンド	34
3.8.5. CaoController::Execute("GetPrinterID") コマンド	35
3.8.6. CaoController::Execute("SetBasePitch") コマンド	36
3.9. 独自コマンド	37
3.9.1. CaoController::Execute("Send") コマンド	37
3.9.2. CaoController::Execute("SendAndReceive") コマンド	37
3.9.3. CaoController::Execute("AddPrint") コマンド	37
3.9.4. CaoController::Execute("AddPrint_JP") コマンド	38
3.9.5. CaoController::Execute("PrintLF_JP") コマンド	39

3.9.6. CaoController::Execute("AddPrint_EN") コマンド.....	39
3.9.7. CaoController::Execute("PrintLF_EN") コマンド.....	40
3.9.8. CaoController::Execute("AddPrint_DE") コマンド.....	40
3.9.9. CaoController::Execute("PrintLF_DE") コマンド.....	41
4. 変数一覧.....	42
4.1.1. システム変数.....	42
5. エラーコード.....	43

1. はじめに

EPSON ESCPOS プロバイダ(以下 ESCPOS プロバイダ)は, ESC/POS¹コマンドを通じてプリンターの制御を行う ORiN2 CAO プロバイダです. ESC/POS コマンドに対応している機種であれば, メーカーを問わず制御することができます.²

本ドキュメントでは, ESCPOS プロバイダの概要と, 実装されている CAO インタフェース(関数仕様)について説明しています.

¹ ESC/POS は, セイコーエプソン株式会社の登録商標です.

² 動作確認には FP-32L-LAN を使用しております.

2. プロバイダの概要

2.1. 概要

ESCPOS プロバイダは、コマンドの実行方法として `CaoController::Execute` による方法を提供しています。`CaoController::Execute` では、Ethernet やシリアルインターフェースを利用して ESC/POS コマンドの送受信を行います。

2.2. インストール

ESCPOS プロバイダモジュールは、下記の DLL で構成されています。ORiN2 SDK のインストーラでインストールした場合は、インストール作業は不要です。手動でインストールする場合は、表 2-1 のように実行してください。

表 2-1 ESCPOS プロバイダ

ファイル名	CaoProv.EPSON.ESCPOS.dll
ProgID	CaoProv. EPSON.ESCPOS
レジストリ登録	regsvr32 CaoProvEPSON.ESCPOS.dll
レジストリ登録の抹消	regsvr32 /u CaoProvEPSON.ESCPOS.dll

2.3. メソッド・プロパティ

2.3.1. CaoWorkspace::AddController メソッド

ESCPOS プロバイダでは AddController 時に、通信用の接続パラメータを参照し、通信の接続を行います。このときオプションで通信形態、タイムアウトを指定します。

書式 AddController(<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])

bstrCtrlName : [in] コントローラ名 任意
 bstrProvName : [in] プロバイダ名 固定値 =" CaoProv.EPSON.ESCPOS"
 bstrPcName : [in] プロバイダの実行マシン名
 bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 2-2 CaoWorkspace::AddController のオプション文字列

オプション	意味
Conn =<接続パラメータ>	必須。 通信形態とその接続パラメータを設定します。 詳細は 2.3.1.1 を参照してください。
MyIP=[<ローカル IP アドレス>[:ローカルポート番号]]	任意。 複数の NIC を使う場合にこのオプションで IP アドレスを指定して NIC を選択することができます。省略した場合は、自動的に選択されます。ローカルマシンに割り当てられていない IP アドレスを指定したときはエラーを返します。
Timeout[=<タイムアウト時間>]	任意。 送受信時のタイムアウト時間を指定します。(ミリ秒) (デフォルト:3000)
ConnTimeout[=<タイムアウト時間>]	任意。 TCP 接続時のタイムアウト時間。(ミリ秒) (デフォルト:500)
KeepAlive[=<キープアライブ時間>]	任意。 TCP 通信時のキープアライブ時間を設定します。(ミリ秒) (デフォルト:1000)

2.3.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで角括弧("[]")内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値を示します。

Ethernet デバイス

"Conn=ETH:<IP Address> [:Port No]"

"Conn=UDP:<IP Address> [:Port No]"³

<IP Address> : 接続する機器の IP アドレス。

例: "127.0.0.1", "192.168.0.1"

<Port No> : 接続ポート番号. 9100, 9101, . . . 任意指定可能

RS-232C デバイス

"Conn=COM:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"³

<COM Port> : COM ポート番号. '1'-COM1, '2'-COM2, . . .

<BaudRate> : 通信速度. 2400, 4800, 9600, 19200, 38400, 57600, 115200.

<Parity> : パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD

<DataBits> : データビット数. '7'-7bit, '8'-8bit.

<StopBits> : ストップビット数. '1'-1bit, '2'-2bit.

<Flow> : フロー制御. '1'-Xon/Xoff, '2'-ハードウェア制御.

OR をとって指定できます。

(デフォルト: '0'-フロー制御なし)

2.3.2. GaoController::Execute メソッド

GaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。第 1 引数にコマンド名、第 2 引数にコマンドのパラメータを指定します。各コマンドの詳細は 3 章を参照してください。

書式 Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

bstrCommandName:[in] コマンド名

vntParam : [in] パラメータ

2.3.3. GaoController::AddVariable メソッド

GaoController クラスの AddVariable メソッドは、変数オブジェクトを作成するためのメソッドです。使用できる変数一覧は 4 章を参照してください。

書式 AddVariable (<bstrCommandName:VT_BSTR>,[<bstrOption : VT_BSTR>])

bstrVariableName : [in] 変数名

³ UDP 通信と RS-232C 通信に関しては接続できるところまでは確認しておりますが、実機での動作確認は行っておりませんのでサポート範囲外となります。

bstrOption : [in] オプション文字列

3. コマンド一覧

本章では CaoController::Execute メソッドの各コマンドについて解説します。以下に ESCPOS プロバイダで対応しているコマンド一覧を示します。

各コマンドの詳細動作についてはコマンドリファレンスを参照してください。

コマンドリファレンスの提供方法は、機種によって異なります。使用される機種をご確認の上、所定の手続きをお願いします。

表 3-1 CaoController::Execute コマンド一覧

ESC/POS コマンド	コマンド	機能	ページ
印字命令コマンド			
LF	PrintLF	印字改行	P. 13
ESC J	PrintFeedPaper	印字および紙送り	P. 13
ESC d	PrintFeedLine	印字および n 行の紙送り	P. 13
改行量コマンド			
ESC 2	ResetLineSpacing	初期改行量に設定	P. 14
ESC 3	SetLineSpacing	改行量の設定	P. 14
印字文字コマンド			
ESC SP	SetRightSideCharacterSpacing	文字の右スペース量の設定	P. 14
ESC !	SetPrintMode	印字モードの一括指定	P. 15
ESC -	SetUnderLineMode	アンダーラインの指定・解除	P. 15
ESC E	SetEmphasizedMode	強調印字の指定・解除	P. 16
ESC G	SetDoubleStrikeMode	二重印字の指定・解除	P. 16
ESC M	SetFont	文字フォントの選択	P. 17
ESC R	SetInternationalCharacter	国際文字の選択	P. 17
ESC V	SetRightRotate	文字の 90 度右回転の指定・解除	P. 18
ESC t	SetCharacterCodeTable	文字コードテーブルの選択	P. 19
ESC {	SetSModeUpsideDownMode	倒立印字の指定・解除	P. 20
GS !	SetFontSize	文字サイズの指定	P. 21
GS B	SetWhiteBlackReverseMode	白黒反転印字の指定・解除	P. 21
GS b	SetSmoothingMode	スムージングの指定・解除	P. 22
印字用紙コマンド			
FS (L <fn=48>	GetPositioningInfo	位置情報の受信	P. 22
FS (L <fn=65>	FeedPaperToPeelingPosition	剥離位置までの紙送り	P. 23
FS (L <fn=66>	FeedPaperToCuttingPosition	カット位置までの紙送り	P. 24
FS (L <fn=67>	FeedPaperToStartingPosition	頭出し位置までの紙送り	P. 24
印字位置コマンド			

HT	MoveHT	水平タブ	P. 25
ESC \$	MoveA	絶対位置の指定	P. 25
ESC D	SetHTPosition	水平タブ位置の設定	P. 26
ESC ¥	MoveR	相対位置の指定	P. 26
ESC a	SetSModeAlignment	位置揃え	P. 27
GS L	SetSModeLeftMargin	左マージンの指定	P. 27
GS T	MoveSModeBeginningLine	行の先頭への印字位置の移動	P. 28
GS W	SetSModePrintAreaWidth	印字領域幅の設定	P. 28
メカコントロールコマンド			
GS V	CutPaper	用紙のカット	P. 29
漢字制御コマンド			
FS !	SetKanjiSetting	漢字の印字モードの一括指定	P. 30
FS &	StartKanjiMode	漢字モードの開始	P. 30
FS -	SetKanjiUnderLineMode	漢字アンダーラインの指定・解除	P. 31
FS .	StopKanjiMode	漢字モードの終了	P. 31
FS C	SetKanjiCodeSystem	漢字コード体系の設定	P. 31
FS S	SetKanjiSpacing	漢字スペース量の設定	P. 32
FS W	SetKanjiQuadrupleSizeMode	漢字の 4 倍角文字の指定・解除	P. 32
FS (A <fn=48>	SetKanjiFont	漢字フォントの選択	P. 33
補助機能コマンド			
DLE DC4 <fn=2>	RTMPowerOff	電源オフ処理の実行	P. 33
DLE DC4 <fn=8>	RTMClearBuffer	バッファークリア	P. 34
ESC @	InitializePrinter	プリンターの初期化	P. 34
GS (A	TestPrint	テスト印字の実行	P. 34
GS I	GetPrinterID	プリンターID の取得	P. 35
GS P	SetBasePitch	基本計算ピッチの設定	P. 36
独自コマンド			
—	Send	指定したコマンドフォーマットを送信します	P. 37
—	SendAndReceive	指定したコマンドフォーマットを送信し、レスポンスを受信します	P. 37
—	AddPrint	プリントバッファに文字列を格納	P. 37
—	AddPrint_JP	プリントバッファに日本語文字列を格納	P. 38
—	PrintLF_JP	日本語印字改行	P. 39
—	AddPrint_EN	プリントバッファに英語文字列を格納	P. 39
—	PrintLF_EN	英語印字改行	P. 40
—	AddPrint_DE	プリントバッファにドイツ語文字列を格納	P. 40
—	PrintLF_DE	ドイツ語印字改行	P. 41

3.1. 印字命令コマンド

3.1.1. GaoController::Execute("PrintLF") コマンド

プリントバッファ内のデータの印字と1行の紙送りを実行します。

パラメータには印字内容を指定することができ、バイト配列か文字列で指定してください。

書式 1 PrintLF ([<vntDataArray>])

<vntDataArray> : [in] 印字内容(バイト配列指定)(VT_UI1 | VT_ARRAY)
戻り値 : なし

書式 2 PrintLF ([<bstrData>])

<bstrData> : [in] 印字内容(文字列指定)(VT_BSTR)
戻り値 : なし

使用例

```
Dim vntDataArray As Variant  
  
vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345  
ctrl.Execute "PrintLF", vntDataArray  
ctrl.Execute "PrintLF", "12345"
```

3.1.2. GaoController::Execute("PrintFeedPaper") コマンド

プリントバッファ内のデータの印字と[byteLine × 基本計算ピッチ]の紙送りを実行します。

書式 PrintFeedPaper (<byteLine>)

<byteLine> : [in] 紙送り量(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "PrintFeedPaper", 100
```

3.1.3. GaoController::Execute("PrintFeedLine") コマンド

プリントバッファ内のデータの印字と byteLine 行の紙送りを実行します。

書式 PrintFeedPaper (<byteLine>)

<byteLine> : [in] 紙送り行数(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "PrintFeedLine", 1
```

3.2. 改行量コマンド

3.2.1. CaoController::Execute("ResetLineSpacing") コマンド

改行量を「初期改行量」に設定します。

書式 ResetLineSpacing ()

引数 : なし
戻り値 : なし

使用例

```
ctrl.Execute "SelectDefaultLineSpacing"
```

3.2.2. CaoController::Execute("SetLineSpacing") コマンド

改行量を[byteData × 基本計算ピッチ]設定します。

書式 SetLineSpacing (<byteData>)

<byteData> : [in] 改行量(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetLineSpacing", 100
```

3.3. 印字文字コマンド

3.3.1. CaoController::Execute("SetRightSideCharacterSpacing") コマンド

文字の右スペース量を[byteData × 基本計算ピッチ]に設定します。

書式 SetRightSideCharacterSpacing (<byteData>)

<byteData> : [in] 右スペース量(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "SetRightSideCharacterSpacing", 100
```

3.3.2. CaoController::Execute("SetPrintMode") コマンド

印字モードを一括指定します。

byteData :ビット	2進数	機能	16進数	10進数
0	0	フォント1を選択する	00	0
	1	フォント2を選択する	01	1
1	-	(予約)	-	-
2	-	(予約)	-	-
3	0	強調印字を解除する	00	0
	1	強調印字を指定する	08	8
4	0	縦倍拡大を解除する	00	0
	1	縦倍拡大を指定する	10	16
5	0	横倍拡大を指定する	00	0
	1	横倍拡大を解除する	20	32
6	-	(予約)	-	-
7	0	アンダーラインの付加を解除する	00	0
8	1	アンダーラインの付加を指定する	80	128

書式

SetPrintMode (<byteData>)

<byteData> : [in] 印字モード(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "SetPrintMode", 1
```

3.3.3. CaoController::Execute("SetUnderLineMode") コマンド

アンダーラインを指定・解除します。

<i>byteData</i>	機能
0, 48	アンダーラインの付加を解除する
1, 49	1ドット幅のアンダーラインの設定とアンダーラインの付加を指定する
2, 50	2ドット幅のアンダーラインの設定とアンダーラインの付加を指定する

書式 SetUnderLineMode (<byteData>)

<byteData> : [in] アンダーライン設定(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetUnderLineMode", 1
```

3.3.4. CaoController::Execute("SetEmphasizedMode") コマンド

強調印字を指定・解除します。

byteData の最下位ビットが 0 のとき、強調印字を解除します。

byteData の最下位ビットが 1 のとき、強調印字を指定します。

書式 SetEmphasizedMode (<byteData>)

<byteData> : [in] 強調印字設定(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetEmphasizedMode", 1
```

3.3.5. CaoController::Execute("SetDoubleStrikeMode") コマンド

二重印字を指定・解除します。

byteData の最下位ビットが 0 のとき、二重印字を解除します。

byteData の最下位ビットが 1 のとき、二重印字を指定します。

書式 SetDoubleStrikeMode (<byteData>)

<byteData> : [in] 二重印字設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetDoubleStrikeMode", 1
```

3.3.6. CaoController::Execute("SetFont") コマンド

文字フォントを選択します。

byteData	フォント
0, 48	フォント A
1, 49	フォント B
2, 50	フォント C
3, 51	フォント D
4, 52	フォント E
97	スペシャルフォント A
98	スペシャルフォント B

書式

SetFont (<byteData>)

<byteData> : [in] 文字フォント(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetFont", 1
```

3.3.7. CaoController::Execute("SetInternationalCharacter") コマンド

国際文字を選択します。

byteData	国名
0	アメリカ
1	フランス
2	ドイツ
3	イギリス
4	デンマーク I
5	スウェーデン
6	イタリア
7	スペイン I

8	日本
9	ノルウェー
10	デンマークII
11	スペインII
12	ラテンアメリカ
13	韓国
14	スロベニア / クロアチア
15	中国
16	ベトナム
17	アラビア
<i>byteData</i>	国名(言語)
66	インド (Devanagari)
67	インド (Bengali)
68	インド (Tamil)
69	インド (Telugu)
70	インド (Assamese)
71	インド (Oriya)
72	インド (Kannada)
73	インド (Malayalam)
74	インド (Gujarati)
75	インド (Punjabi)
82	インド (Marathi)

書式 SetInternationalCharacter (<byteData>)

<byteData> : [in] 各国の文字セット(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetInternationalCharacter", 0
```

3.3.8. CaoController::Execute("SetRightRotate") コマンド

文字の 90 度右回線の指定・解除します。

<i>byteData</i>	機能
0, 48	90 度右回転文字を解除する
1, 49	90 度右回転文字 (横方法 1 ドット間隔) を指定する
2, 50	90 度右回転文字 (横方向 1.5 ドット間隔) を指定する

書式 SetRightRotate (<byteData>)

<byteData> : [in] 文字の回転指定(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetRightRotate", 1
```

3.3.9. CaoController::Execute("SetCharacterCodeTable") コマンド

文字コードテーブルを選択します。

byteData	選択ページ[文字種類]
0	ページ 0 [PC437: USA, Standard Europe]
1	ページ 1 [カタカナ]
2	ページ 2 [PC850: Multilingual]
3	ページ 3 [PC860: Portuguese]
4	ページ 4 [PC863: Canadian-French]
5	ページ 5 [PC865: Nordic]
6	ページ 6 [ひらがな・簡易漢字]
7	ページ 7 [簡易漢字]
8	ページ 8 [簡易漢字]
11	ページ 11 [PC851: Greek]
12	ページ 12 [PC853: Turkish]
13	ページ 13 [PC857: Turkish]
14	ページ 14 [PC737: Greek]
15	ページ 15 [ISO8859-7: Greek]
16	ページ 16 [WPC1252]
17	ページ 17 [PC866: Cyrillic #2]
18	ページ 18 [PC852: Latin 2]
19	ページ 19 [PC858: Euro]
20	ページ 20 [Thai Character Code 42]
21	ページ 21 [Thai Character Code 11]
22	ページ 22 [Thai Character Code 13]
23	ページ 23 [Thai Character Code 14]
24	ページ 24 [Thai Character Code 16]
25	ページ 25 [Thai Character Code 17]
26	ページ 26 [Thai Character Code 18]
30	ページ 30 [TCVN-3: Vietnamese]
31	ページ 31 [TCVN-3: Vietnamese]
32	ページ 32 [PC720: Arabic]
33	ページ 33 [WPC775: Baltic Rim]
34	ページ 34 [PC855: Cyrillic]
35	ページ 35 [PC861: Icelandic]
36	ページ 36 [PC862: Hebrew]

37	ページ 37 [PC864: Arabic]
38	ページ 38 [PC869: Greek]
39	ページ 39 [ISO8859-2: Latin 2]
40	ページ 40 [ISO8859-15: Latin 9]
41	ページ 41 [PC1098: Farsi]
42	ページ 42 [PC1118: Lithuanian]
43	ページ 43 [PC1119: Lithuanian]
44	ページ 44 [PC1125: Ukrainian]
45	ページ 45 [WPC1250: Latin 2]
46	ページ 46 [WPC1251: Cyrillic]
47	ページ 47 [WPC1253: Greek]
48	ページ 48 [WPC1254: Turkish]
49	ページ 49 [WPC1255: Hebrew]
50	ページ 50 [WPC1256: Arabic]
51	ページ 51 [WPC1257: Baltic Rim]
52	ページ 52 [WPC1258: Vietnamese]
53	ページ 53 [KZ-1048: Kazakhstan]
66	ページ 66 [Devanagari]
67	ページ 67 [Bengali]
68	ページ 68 [Tamil]
69	ページ 69 [Telugu]
70	ページ 70 [Assamese]
71	ページ 71 [Oriya]
72	ページ 72 [Kannada]
73	ページ 73 [Malayalam]
74	ページ 74 [Gujarati]
75	ページ 75 [Punjabi]
82	ページ 82 [Marathi]
254	ページ 254
255	ページ 255

書式 SetCharacterCodeTable (<byteData>)

<byteData> : [in] 文字コード表のページ番号(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetCharacterCodeTable", 0
```

3.3.10. GaoController::Execute("SetSMModeUpsideDownMode") コマンド

倒立印字を指定・解除します。

byteData の最下位ビットが 0 のとき、倒立印字を解除します。

byteData の最下位ビットが 1 のとき、倒立印字を指定します。

書式 SetSMModeUpsideDownMode (<byteData>)

<byteData> : [in] 倒立印字設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetSMModeUpsideDownMode", 1
```

3.3.11. CaoController::Execute("SetFontSize") コマンド

文字サイズを設定します。

byteData のビット 6, 5, 4 で文字の横方向倍率を指定し、ビット 2, 1, 0 で文字の縦方向倍率を指定します (0xxx0xxx**b**)。

縦方向倍率の指定

ビット			16 進数	10 進数	倍率
6	5	4			
0	0	0	00	0	1 倍 (標準)
0	0	1	10	16	2 倍 (横倍)
0	1	0	20	32	3 倍
0	1	1	30	48	4 倍
1	0	0	40	64	5 倍
1	0	1	50	80	6 倍
1	1	0	60	96	7 倍
1	1	1	70	112	8 倍

横方向倍率の指定

ビット			16 進数	10 進数	倍率
6	5	4			
0	0	0	00	0	1 倍 (標準)
0	0	1	01	1	2 倍 (縦倍)
0	1	0	02	2	3 倍
0	1	1	03	3	4 倍
1	0	0	04	4	5 倍
1	0	1	05	5	6 倍
1	1	0	06	6	7 倍
1	1	1	07	7	8 倍

書式 SetFontSize (<byteData>)

<byteData> : [in] 文字サイズ設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetFontSize", 18 ' 縦 2 倍横 3 倍
```

3.3.12. CaoController::Execute("SetWhiteBlackReverseMode") コマンド

白黒反転印字を指定・解除します。

byteData の最下位ビットが 0 のとき、白黒反転印字を解除します。

byteData の最下位ビットが 1 のとき、白黒反転印字を指定します。

書式 `SetWhiteBlackReverseMode (<byteData>)`

<byteData> : [in] 白黒反転印字設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetWhiteBlackReverseMode", 1
```

3.3.13. CaoController::Execute("SetSmoothingMode") コマンド

スムージングを指定・解除します。

byteData の最下位ビットが 0 のとき、スムージングを解除します。

byteData の最下位ビットが 1 のとき、スムージングを指定します。

書式 `SetSmoothingMode (<byteData>)`

<byteData> : [in] スムージング設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetSmoothingMode", 1
```

3.4. 印字用紙コマンド

3.4.1. CaoController::Execute("GetPositioningInfo") コマンド

ラベル紙およびブラックマーク紙の位置情報を受信します。

・[位置情報 A]は次表となります。

ビット	2進数	機能	16進数	10進数
0	0	剥離位置でない	00	0
	1	剥離位置に待機	01	1
1	0	カット位置でない	00	0
	1	カット位置に待機	02	2
2	0	頭出し位置でない	00	0

	1	頭出し位置に待機	04	4
3 ~ 5	-	(予約)	-	-
6	1	固定	40	64
7	0	固定	00	0

•[位置情報 B]は次表となります。

ビット	2進数	機能	16進数	10進数
0	0	現在ラベルの頭出し動作が可能	00	0
	1	現在ラベルの頭出し動作が不可能	01	1
1	0	次ラベルの頭出し動作が可能	00	0
	1	次ラベルの頭出し動作が不可能	02	2
2 ~ 5	-	(予約)	-	-
6	1	固定	40	64
7	0	固定	00	0

書式

GetPositioningInfo ()

引数 : なし
 戻り値 : [out] 位置情報を返します (VT_UI1 | VT_ARRAY)
 0:位置情報 A
 1:位置情報 B

使用例

```
Dim vntResult as Variant
vntResult = ctrl.Execute "GetPositioningInfo"
```

3.4.2. CaoController::Execute("FeedPaperToPeelingPosition") コマンド

剥離位置まで紙送りします。

byteData	機能
48	次の剥離位置までの紙送りを実行する。 ただし、剥離位置に待機している場合は動作しない。
49	次の剥離位置までの紙送りを実行する。 ただし、剥離位置に待機している場合は更に次の剥離位置までの紙送りを実行する。

書式

FeedPaperPeelingPosition (<byteData>)

<byteData> : [in] 紙送りモード(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "FeedPaperPeelingPosition", 48
```

3.4.3. CaoController::Execute("FeedPaperToCuttingPosition") コマンド

カット位置まで紙送りします。

byteData	機能
48	次のカット位置までの紙送りを実行する。 ただし、カット位置に待機している場合は動作しない。
49	次のカット位置までの紙送りを実行する。 ただし、カット位置に待機している場合は更に次のカット位置までの紙送りを実行する。

書式

FeedPaperToCuttingPosition (<byteData>)

<byteData> : [in] 紙送りモード(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "FeedPaperToCuttingPosition", 48
```

3.4.4. CaoController::Execute("FeedPaperToStartingPosition") コマンド

頭出し位置まで紙送りします。

byteData	機能
48	次ラベルの頭出し位置までの紙送りを実行する。 ただし、頭出し位置に待機している場合は動作しない。
49	次の頭出し位置までの紙送りを実行する。 ただし、頭出し位置に待機している場合は更に次の頭出し位置までの紙送りを実行する。
50	現在ラベルの頭出し位置までの紙送りを実行する。 ただし、頭出し位置に待機している場合は動作しない。

書式

FeedPaperStartingPosition (<byteData>)

<byteData> : [in] 紙送りモード(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "FeedPaperStartingPosition", 48
```

3.5. 印字位置コマンド

3.5.1. GaoController::Execute("MoveHT") コマンド

印字位置を次の水平タブまで移動します。

書式 MoveHT ()

引数 : なし
 戻り値 : なし

使用例

```
ctrl.Execute "MoveHT"
```

3.5.2. GaoController::Execute("MoveA") コマンド

印字位置を印字領域の左端から[(byteNL + byteNH × 256) × 基本計算ピッチ]の位置へ移動します。

書式 1 MoveA (<byteNL>, <byteNH>)

<byteNL> : [in] 絶対位置(VT_UI1)
 <byteNH> : [in] 絶対位置(VT_UI1)
 戻り値 : なし

書式 2 MoveA (<wordPOS>)

<wordPOS> : [in] 絶対位置(VT_UI2)
 $\text{wordPOS} = (\text{byteNL} + \text{byteNH} \times 256)$
 戻り値 : なし

使用例

```
ctrl.Execute "MoveA", Array(1, 2)
```

ctrl.Execute "MoveA", 513

3.5.3. CaoController::Execute("SetHTPosition") コマンド

水平タブ位置を設定します。

vntDataArray は印字領域の左端から水平タブ位置までの桁数を指定します。

設定可能な水平タブ数は 32 個までです。

水平タブ位置配列を設定しない(VT_EMPTY または VT_NULL)場合は設定されている水平タブをクリアします。

書式 1 SetHTPosition (<vntDataArray>)

<vntDataArray> : [in] 水平タブ位置配列(VT_UI1 | VT_ARRAY)
戻り値 : なし

書式 2 SetHTPosition ()

引数 : なし
戻り値 : なし

使用例

```
ctrl.Execute "SetHTPosition", Array(8, 16, 24, 32)  
ctrl.Execute "SetHTPosition"
```

3.5.4. CaoController::Execute("MoveR") コマンド

印字位置を現在位置から[(*byteNL* + *byteNH* × 256) × 基本計算ピッチ]の位置へ移動します。

現在位置より右方向へ移動する場合は正数(プラス)を, 左方向へ移動する場合は負数(マイナス)を指定します。

書式 1 MoveR (<byteNL>, <byteNH>)

<byteNL> : [in] 相対位置(VT_UI1)
<byteNH> : [in] 相対位置(VT_UI1)
戻り値 : なし

書式 2 MoveR (<shortPos>)

<shortPos> : [in] 相対位置(VT_I2)

shortPos = (byteNL + byteNH × 256)

戻り値 : なし

使用例

```
ctrl.Execute "MoveR", Array(1, 2)
ctrl.Execute "MoveR", 513
ctrl.Execute "MoveR", -513
```

3.5.5. GaoController::Execute("SetSMModeAlignment") コマンド

印字データの配置を設定します。

byteData	機能
0, 48	「左揃え」に設定する
1, 49	「中央揃え」に設定する
2, 50	「右揃え」に設定する

書式

SetSMModeAlignment (<byteData>)

<byteData> : [in] 配置位置(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetSMModeAlignment", 1
```

3.5.6. GaoController::Execute("SetSMModeLeftMargin") コマンド

左マージンを印字可能領域の左端から[(byteNL + byteNH × 256) × 基本計算ピッチ]に設定します。

書式 1

SetSMModeLeftMargin (<byteNL>, <byteNH>)

<byteNL> : [in] 左マージン位置(VT_UI1)
<byteNH> : [in] 左マージン位置(VT_UI1)
戻り値 : なし

書式 2

SetSMModeLeftMargin (<wordPos>)

<wordPos> : [in] 左マージン位置(VT_UI2)

wordPos = (byteNL + byteNH × 256)

戻り値 : なし

使用例

```
ctrl.Execute "SetSMModeLeftMargin", Array(1, 2)
ctrl.Execute "SetSMModeLeftMargin", 513
```

3.5.7. CaoController::Execute("MoveSMModeBeginningLine") コマンド

指定動作の実行後にプリンターを「行の先頭」の状態にし、印字位置を行の先頭へ移動します。

byteData	機能
0, 48	現在のプリントバッファ内のデータを消去する
1, 49	現在のプリントバッファ内のデータを印字する

書式

MoveSMModeBeginningLine (<byteData>)

<byteData> : [in] 操作モード(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "MoveSMModeBeginningLine", 0
```

3.5.8. CaoController::Execute("SetSMModePrintAreaWidth") コマンド

印字領域幅を[(byteNL + byteNH × 256) × 基本計算ピッチ]に設定します。

書式 1

SetSMModePrintAreaWidth (<byteNL>, <byteNH>)

<byteNL> : [in] 印字領域幅(VT_UI1)
<byteNH> : [in] 印字領域幅(VT_UI1)
戻り値 : なし

書式 2

SetSMModePrintAreaWidth (<wordPOS>)

<wordPos> : [in] 印字領域幅(VT_UI2)
wordPos = (byteNL + byteNH × 256)

戻り値 : なし

使用例

```
ctrl.Execute "SetSModePrintAreaWidth", Array(1, 2)
ctrl.Execute "SetSModePrintAreaWidth", 513
```

3.6. メカコントロールコマンド

3.6.1. GaoController::Execute("CutPaper") コマンド

用紙のカットを実行します。

vntMode で指定された用紙のカットを実行します。

機能名	機能	<i>vntMode</i>	カット形状
<機能 A>	用紙のカット	0, 48	フルカット
		1, 49	パーシャルカット
<機能 B>	紙送りと用紙のカット	65	フルカット
		66	パーシャルカット
<機能 C>	用紙カット位置の予約	97	フルカット
		98	パーシャルカット
<機能 D>	紙送りと用紙のカットと用紙の頭出し	103	フルカット
		104	パーシャルカット

各機能動作は以下の通りです。

機能名	機能
<機能 A>	用紙のカットを実行します。
<機能 B>	[カット位置 + (<i>vntPos</i> × 縦方向基本計算ピッチ)]まで紙送り後、用紙をカットします。
<機能 C>	[カット位置 + (<i>vntPos</i> × 縦方向基本計算ピッチ)]を用紙カット位置として設定し、設定位置が印字・紙送りなどでオートカッター位置に来たとき、用紙をカットします。
<機能 D>	[カット位置 + (<i>vntPos</i> × 縦方向計算ピッチ)]までの紙送りと用紙のカットを実行した後、バックフィードにより用紙の頭出しをおこないます。

書式 CutPaper (<*vntMode*> [, <*vntPos*>])

<*vntMode*> : [in] カットモード(VT_UI1)
 <*vntPos*> : [in] カット位置(VT_UI1)
 戻り値 : なし

使用例

```
Dim vntDataArray As Variant
```

```
vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345
ctrl.Execute "PrintLF", vntDataArray
ctrl.Execute "PrintLF", "12345"
```

3.7. 漢字制御コマンド

3.7.1. CaoController::Execute("SetKanjiSetting") コマンド

マルチバイトコード文字の文字装飾を一括指定します。

<i>byteData</i> :ビット	2進数	機能	16進数	10進数
0	0	(予約)	00	0
1	0	(予約)	00	0
2	0	横倍拡大を解除する	00	0
	1	横倍拡大を指定する	04	4
3	0	縦倍拡大を解除する	00	0
	1	縦倍拡大を指定する	08	8
4 ~ 6	0	(予約)	00	0
7	0	漢字アンダーラインの付加を解除する	00	00
	1	漢字アンダーラインの付加を指定する	80	128

書式

SetKanjiSetting (<byteData>)

<byteData> : [in] 漢字モード(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiSetting", 4
```

3.7.2. CaoController::Execute("StartKanjiMode") コマンド

漢字モードを開始します。

書式

StartKanjiMode ()

引数 : なし

戻り値 : なし

使用例

```
ctrl.Execute "StartKanjiMode"
```

3.7.3. CaoController::Execute("SetKanjiUnderLineMode") コマンド

漢字アンダーラインを指定・解除します。

byteData	機能
0, 48	漢字アンダーラインの付加を解除する
1, 49	1ドット幅の漢字アンダーラインの設定とアンダーラインの付加を指定する
2, 50	2ドット幅の漢字アンダーラインの設定とアンダーラインの付加を指定する

書式

SetKanjiUnderLineMode (<byteData>)

<byteData> : [in] 漢字アンダーライン設定(VT_UII)

戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiUnderLineMode", 1
```

3.7.4. CaoController::Execute("StopKanjiMode") コマンド

漢字モードを終了します。

書式

StopKanjiMode ()

引数 : なし

戻り値 : なし

使用例

```
ctrl.Execute "StopKanjiMode"
```

3.7.5. CaoController::Execute("SetKanjiCodeSystem") コマンド

日本語仕様における JIS コード体系, またはシフト JIS コード体系を選択します。

<i>byteData</i>	機能
0, 48	JIS コード体系を選択する
1, 49	シフト JIS コード体系を選択する

書式 SetKanjiCodeSystem (<byteData>)

<byteData> : [in] 漢字コード体系(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiCodeSystem", 1
```

3.7.6. CaoController::Execute("SetKanjiSpacing") コマンド

マルチバイトコード文字の左スペース量および右スペース量を設定します。

左スペース量を[*byteLSpace* × 基本計算ピッチ]に設定します。

右スペース量を[*byteRSpace* × 基本計算ピッチ]に設定します。

書式 SetKanjiSpacing (<byteLSpace>, <byteRSpace>)

<byteLSpace> : [in] 左スペース量(VT_UI1)
<byteRSpace> : [in] 右スペース量(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiSpacing", Array(1, 2)
```

3.7.7. CaoController::Execute("SetKanjiQuadrupleSizeMode") コマンド

マルチバイトコード文字の 4 倍角文字を指定・解除します。

byteData の最下位ビットが 0 のとき, 4 倍角文字を解除し標準サイズを指定します。

byteData の最下位ビットが 1 のとき, 4 倍角文字を指定します。

書式 SetKanjiQuadrupleSizeMode (<byteData>)

<byteData> : [in] 4 倍角文字設定(VT_UI1)
戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiQuadrupleSizeMode", 1
```

3.7.8. CaoController::Execute("SetKanjiFont") コマンド

マルチバイトコード文字のフォント(漢字フォント)を選択します。

<i>byteData</i>	機能
0, 48	漢字フォント A を選択する
1, 49	漢字フォント B を選択する
2, 50	漢字フォント C を選択する

書式

SetKanjiFont (<byteData>)

<byteData> : [in] 漢字フォント設定(VT_UI1)

戻り値 : なし

使用例

```
ctrl.Execute "SetKanjiFont", 0
```

3.8. 補助機能コマンド**3.8.1. CaoController::Execute("RTMPowerOff") コマンド**

電源オフの準備処理を実行し、電源 OFF 通知を受信します。

本コマンドは電源 OFF 通知を正常に受信した段階で処理が終了します。

何らかの原因により電源 OFF 通知を受信できなかった場合にはエラーが発生します。

書式

RTMPowerOff ()

引数 : なし

戻り値 : なし

使用例

```
ctrl.Execute "RTMPowerOff"
```

3.8.2. CaoController::Execute("RTMClearBuffer") コマンド

受信バッファおよびプリントバッファ内の全データを消去した後、クリアレスポンスを受信します。
本コマンドはクリアレスポンスを正常に受信した段階で処理が終了します。
何らかの原因によりクリアレスポンスを受信できなかった場合にはエラーが発生します。

書式 RTMClearBuffer ()

引数 : なし
戻り値 : なし

使用例

```
Dim vntResult as Variant
vntResult = ctrl.Execute "RTMClearBuffer"
```

3.8.3. CaoController::Execute("InitializePrinter") コマンド

プリントバッファのクリアと、一部コマンド⁴の設定値を初期値に戻します。
本コマンドは AddController 時にも内部実行されます。

書式 InitializePrinter ()

引数 : なし
戻り値 : なし

使用例

```
ctrl.Execute "InitializePrinter"
```

3.8.4. CaoController::Execute("TestPrint") コマンド

指定されたテスト印字を実行します。

bytePaper は次表に示すテスト印字の対象となる用紙を指定します。

<i>byteData</i>	機能
0, 48	基本シート(ロール紙)
1, 49 2, 50	ロール紙

byteTestPattern は次表に示すテスト印字の種類を指定します。

⁴ 初期値に戻るコマンドの内容に関してはコマンドリファレンスを参照してください

<i>byteTestPattern</i>	機能
1, 49	16 進ダンプ
2, 50	プリンターの状態印字
3, 51	ローリングパターン印字
64	用紙レイアウトの自動設定

書式 TestPrint (<bytePaper>, <byteTestPattern>)

<bytePaper> : [in] 用紙の種類(VT_UI1)
 <byteTestPattern> : [in] テスト印字の種類(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "TestPrint", Array(1, 1)
```

3.8.5. CaoController::Execute("GetPrinterID") コマンド

指定されたプリンターID またはプリンター情報を受信します。

・指定されたプリンターID を受信します。

<i>byteData</i>	プリンターID	仕様
1, 49	機種 ID	機種
2, 50	タイプ ID	サポートしている機能
3, 51	バージョン ID	ファームウェアのバージョン

・指定されたプリンター情報 A を受信します。

<i>byteData</i>	プリンター情報の種類	仕様
33	タイプ情報	サポートしている機能
35, 36, 96, 110	[機種情報]を参照ください	[機種情報]を参照してください

・指定されたプリンター情報 B を受信します。

<i>byteData</i>	プリンター情報の種類	仕様
65	ファームウェアバージョン	ファームウェアのバージョン
66	メーカー名	メーカー名
67	機種名	機種名
68	シリアル番号	製品のシリアル番号
69	多国語フォントの種類	日本語仕様:"KANJI JAPANESE"
		簡体字中国語仕様:"CHINA GB18030"または "CHINA GB2312"
		繁体字中国語仕様:"TAIWAN BIG-5"

		韓国語仕様:"KOREA C-5601C"
		南アジア仕様:"THAI 1 PASS"
111	[機種情報]を参照してください	[機種情報]を参照してください
112	[機種情報]を参照してください	[機種情報]を参照してください

書式

GetPrinterID (<byteData>)

<byteData> : [in] 取得番号(VT_UI1)
 戻り値 : [out] 受信データ
 [取得番号=1~3, 49~51]
 プリンターID (VT_UI1)
 [取得番号=33, 35, 96, 110]
 プリンター情報 A (VT_UI1 | VT_ARRAY)
 [取得番号=65~69]
 プリンター情報 B (VT_BSTR)
 [取得番号=111, 112]
 プリンター情報 B (VT_UI | VT_ARRAY)

使用例

```
Dim vntResult as Variant
vntResult = ctrl.Execute "GetPrinterID", 0
```

3.8.6. CaoController::Execute("SetBasePitch") コマンド

横方向の基本計算ピッチを 25.4/x mm {1/x インチ}, 縦方向の基本計算ピッチを 25.4/y mm {1/y インチ} に設定します。

(byteX = 0)の場合は, 横方向の基本計算ピッチを初期値に戻します。

(byteY = 0)の場合は, 縦方向の基本計算ピッチを初期値に戻します。

書式

SetBasePitch (<byteX>, <byteY>)

<byteX> : [in] 横方向基本計算ピッチ(VT_UI1)
 <byteY> : [in] 縦方向基本計算ピッチ(VT_UI1)
 戻り値 : なし

使用例

```
ctrl.Execute "SetBasePitch", Array(1, 2)
```

3.9. 独自コマンド

3.9.1. GaoController::Execute ("Send") コマンド

指定したコマンドフォーマットを送信します。

書式 Send (<vntDataArray>)

<vntDataArray> : [in] 送信バイト配列(VT_UI1 | VT_ARRAY)

戻り値 : なし

使用例

```
Dim vntDataArray As Variant  
vntDataArray = Array(&h31, &h32, &h33, &h34, &h35, &h0A) ' 12345LF  
ctrl.Execute "Send", vntDataArray
```

3.9.2. GaoController::Execute ("SendAndReceive") コマンド

指定したコマンドフォーマットを送信し、レスポンスデータを受信します。

応答がないコマンドを実行した場合、データ受信待ちになりタイムアウトエラー(0x80000900)が発生します。

書式 SendAndReceive (<vntDataArray>)

<vntDataArray> : [in] 送信バイト配列(VT_UI1 | VT_ARRAY)

戻り値 : [out] 受信したレスポンスデータを一切加工せずヘッダーも含めて返します
(VT_UI1 | VT_ARRAY)

使用例

```
Dim vntDataArray As Variant  
Dim vntResult as Variant  
vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345  
ctrl.Execute "Send", vntDataArray  
vntDataArray = Array(&h10, &h14, &h08, &h01, &h03, &h14, &h01, &h06, &h02, &h08) ' DLE DC4 fn(8)  
vntResult = ctrl.Execute("SendAndReceive", vntDataArray)
```

3.9.3. GaoController::Execute ("AddPrint") コマンド

プリントバッファに文字列、コマンドデータを格納。

送信するデータがない場合は引数が不正エラー(0x80070057)が発生します。

書式 1 AddPrint (<vntDataArray>)

<vntDataArray> : [in] 送信バイト配列(VT_UI1 | VT_ARRAY)
戻り値 : なし

書式 2 AddPrint (<bstrData>)

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
Dim vntDataArray As Variant  
vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345  
ctrl.Execute "AddPrint", vntDataArray  
ctrl.Execute "AddPrint", "12345"
```

3.9.4. GaoController::Execute("AddPrint_JP") コマンド

プリントバッファに日本語文字列を格納します。

送信するデータがない場合は引数が不正エラー(0x80070057)が発生します。

日本語(ひらがな, カタカナ, 漢字)を印字する際は AddPrint_JP または PrintLF_JP を使用すると便利です。

AddPrint_JP は内部で「国際文字の選択(日本)」、「文字コードテーブルの選択」(ページ 1[カタカナ]), 「漢字コード体系の選択(シフト JIS)」に変更しています。

処理完了後は「国際文字の選択(日本)」、「文字コードテーブルの選択(ページ 0[PC437: USA, StandardEurope])」、「漢字コード体系の選択(JIS)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合はプリントバッファの送信までは成功していますが、戻し処理で異常が発生しているため設定状態は不定となります。

書式 AddPrint_JP (<bstrData>)

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
ctrl.Execute "AddPrint_JP", "12345abcde あいうえお7ㄗㄗアイウエオ垂位鵜絵尾"
```

3.9.5. CaoController::Execute("PrintLF_JP") コマンド

プリントバッファに日本語文字列を格納し印字と1行の紙送りを実行します。

送信するデータがない場合は引数が不正エラー(0x80070057)が発生します。

日本語(ひらがな, カタカナ, 漢字)を印字する際は AddPrint_JP または PrintLF_JP を使用すると便利です。

PrintLF_JP は内部で「国際文字の選択(日本)」、「文字コードテーブルの選択」(ページ 1[カタカナ]), 「漢字コード体系の選択(シフト JIS)」に変更しています。

処理完了後は「国際文字の選択(日本)」、「文字コードテーブルの選択(ページ 0[PC437: USA, StandardEurope])」、「漢字コード体系の選択(JIS)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合は印字処理までは成功していますが, 戻し処理で異常が発生しているため設定状態は不定となります。

書式 PrintLF_JP ([<bstrData>])

<bstrData> : [in] 送信バイト文字列(VT_BSTR)

戻り値 : なし

使用例

```
ctrl.Execute "PrintLF_JP", "12345abcde あいうえお7ㄗㄗアイウエオ垂位鵜絵尾"
```

3.9.6. CaoController::Execute("AddPrint_EN") コマンド

プリントバッファに英語文字列を格納します。

送信するデータがない場合は引数が不正エラー(0x80070057)が発生します。

英語を印字する際は AddPrint_EN または PrintLF_EN を使用すると便利です。

AddPrint_EN は内部で「国際文字の選択(アメリカ)」、「文字コードテーブルの選択」(ページ 0[PC437: USA, StandardEurope])に変更しています。

処理完了後は「国際文字の選択(日本)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合はプリントバッファの送信までは成功していますが, 戻し処理で異常が発生しているため設定状態は不定となります。

書式 AddPrint_EN (<bstrData>)

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
ctrl.Execute "AddPrint_EN", "12345abcde"
```

3.9.7. CaoController::Execute("PrintLF_EN") コマンド

プリントバッファに日本語文字列を格納し印字と1行の紙送りを実行します。
英語を印字する際は AddPrint_EN または PrintLF_EN を使用すると便利です。

PrintLF_EN は内部で「国際文字の選択(アメリカ)」、「文字コードテーブルの選択」(ページ 0[PC437: USA, StandardEurope]))に変更しています。

処理完了後は「国際文字の選択(日本)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合は印字処理までは成功していますが、戻し処理で異常が発生しているため設定状態は不定となります。

書式 PrintLF_EN ([<bstrData>])

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
ctrl.Execute "PrintLF_EN", "12345abcde"
```

3.9.8. CaoController::Execute("AddPrint_DE") コマンド

プリントバッファにドイツ語文字列を格納します。
送信するデータがない場合は引数が不正エラー(0x80070057)が発生します。
ドイツ語を印字する際は AddPrint_DE または PrintLF_DE を使用すると便利です。

AddPrint_DE は内部で「国際文字の選択(ドイツ)」、「文字コードテーブルの選択」(ページ 0[PC437: USA, StandardEurope]))に変更しています。

処理完了後は「国際文字の選択(日本)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合はプリントバッファの送信までは成功していますが、戻し処理で異常が発生しているため設定状態は不定となります。

書式 AddPrint_DE (<bstrData>)

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
ctrl.Execute "AddPrint_DE", "12345abcde[¥] {}"      ' 12345abcdeÄÖÜäöüß
```

3.9.9. CaoController::Execute("PrintLF_DE") コマンド

プリントバッファにドイツ文字列を格納し印字と1行の紙送りを実行します。

ドイツ語を印字する際は AddPrint_DE または PrintLF_DE を使用すると便利です。

PrintLF_DE は内部で「国際文字の選択(ドイツ)」、「文字コードテーブルの選択」(ページ 0[PC437: USA, StandardEurope])に変更しています。

処理完了後は「国際文字の選択(日本)」に処理を戻しています。

送信後処理異常(0x80100002)が発生した場合は印字処理までは成功していますが、戻し処理で異常が発生しているため設定状態は不定となります。

書式 PrintLF_DE ([<bstrData>])

<bstrData> : [in] 送信バイト文字列(VT_BSTR)
戻り値 : なし

使用例

```
ctrl.Execute "PrintLF_DE", "12345abcde[¥] {}"      ' 12345abcdeÄÖÜäöüß
```

4. 変数一覧

4.1.1. システム変数

表 4-1 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名="EPSON"を返す.	○	—
@VERSION	VT_BSTR	プロバイダバージョン情報を返す.	○	—

5. エラーコード

ESCPOS プロバイダでは、以下の固有エラーコードが定義されています。

ORiN2 共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。

表 5-1 固有エラーコード

エラー名	エラー番号	説明
受信パケット異常	0x80100000	受信データフォーマット異常。
特定データ準備異常	0x80100001	機器が特定データを準備できなかった。
送信後処理異常	0x80100002	送信後の後処理で失敗しました。